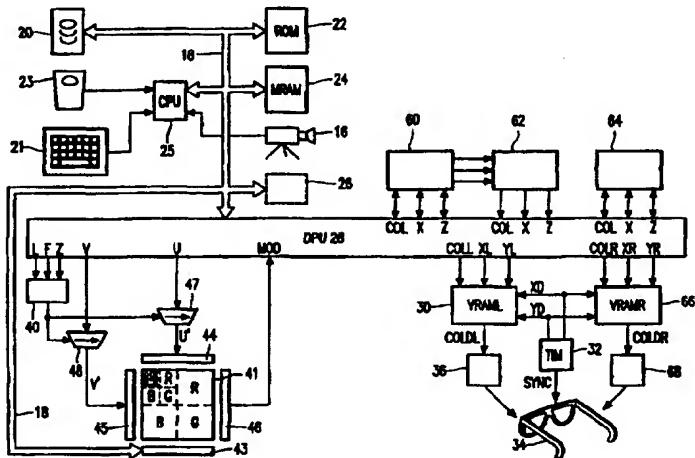




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :  H04N 13/00		A2	(11) International Publication Number: <b>WO 97/47141</b>
			(43) International Publication Date: 11 December 1997 (11.12.97)
(21) International Application Number: PCT/IB97/00588			(81) Designated States: JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 23 May 1997 (23.05.97)			
(30) Priority Data: 9611938.3 7 June 1996 (07.06.96) GB			Published <i>Without international search report and to be republished upon receipt of that report.</i>
(71) Applicant: PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).			
(71) Applicant (for SE only): PHILIPS NORDEN AB [SE/SE]; Kottbygatan 7, Kista, S-164 85 Stockholm (SE).			
(72) Inventor: WOOD, Karl, Joseph; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).			
(74) Agent: WHITE, Andrew, G.; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).			

## (54) Title: STEREO IMAGE GENERATION



## (57) Abstract

A post-processing method and apparatus are described whereby a second image is formed by horizontally shifting pixel primitives of a first image by an amount determined by their respective depths to give a parallax shift. To avoid gaps appearing in the shifted image, two z-buffers (60, 62) are maintained during rendering of the first image with the second (62) being loaded by discarded values of the first (60) such as to provide a record of the first occluded primitive for each pixel. The horizontal shift is applied to the contents of both z-buffers and the results written to a third z-buffer (64) such that gaps due to the varied shift applied to the first image values (the unshifted contents of the first buffer 60) are filled by previously occluded values from the second buffer (62). The two images may then be output to a suitable stereoscopic display such as a Virtual Reality headset (34).

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

## DESCRIPTION

## STEREO IMAGE GENERATION

5 The present invention relates to image generation methods and apparatuses, and in particular to the generation of images by post-processing techniques, from existing image data, for stereoscopic display applications.

10 Stereoscopy in three-dimensional (3-D) graphics refers to techniques for delivering different but correlated images to the left and right eyes in order to stimulate the perception of depth: a number of such techniques are described in "Computer Graphics: Principles and Practice" by James D. Foley et al, 2nd Edition, published by Addison-Wesley 1990, ISBN 0-201-12110-7 at pages 915 to 917. Typically, one 3-D graphics renderer generates the image for the left 15 eye and another generates the image for the right eye. These images might then be fed to a Head Mounted Display or similar stereoscopic display device. There are also multiple view autostereoscopic display devices such as the Philips 3D LCD where, instead of separate images for each of two viewpoints 20 there are in general N images respectively for N viewpoints. Dedicating one 3-D graphics renderer to rendering each image can become expensive, and time multiplexing a renderer is often impractical because such a renderer is typically used to capacity in rendering one image per frame period in order to provide for smooth motion.

25 To avoid such drawbacks, so-called synthetic stereoscopy may be used: in a simple stereoscopic application where respective images are required for the left and right eyes, pixels in the original image can be shifted to the left or right as a function of their distance from the viewer (expressed as their depth) 30 to synthesise left and right images which rely on the parallax introduced by the shifting to stimulate perception of depth. Whilst the technique is cheaper than providing a separate renderer per view, it suffers from gaps created in the images where pixels have been shifted to the left or right. Rendering a central

image and shifting to obtain left and right images creates smaller gaps than rendering for example the left eye image and shifting to generate the right eye image, but the latter requires less resources.

5 It is therefore an object of the present invention to avoid the problems of gaps appearing due to left/right image shifts in stereoscopic applications.

In accordance with the present invention there is provided a method for generating a stereo pair of two-dimensional (2-D) images of an object made up of a plurality of image primitives in three-dimensional (3-D) object space, the 10 first image being from a predetermined viewpoint within the object space and the method comprising the steps of:

a) z-buffering the primitive data to select the nearest opaque primitive per pixel of the first 2-D image and rejecting pixel primitive data having a greater depth;

15 b) z-buffering the rejected pixel primitive data from step (a) to select the first occluded primitive per pixel of the first 2-D image;

c) applying a positional offset in one of said two dimensions to the pixel primitive data selected in steps (a) and (b) in an amount per pixel determined by the respective depth value of that pixel;

20 d) z-buffering the shifted pixel primitive data from step (c) to select preferential contributions per pixel; and

e) rendering the first image from the pixel primitive data selected at step (a) and the second from the pixel primitive data selected at step (d).

Suitably, step (a) may further comprise selecting non-opaque primitives 25 if these are nearer than the said nearest opaque primitive per pixel, and generating a per pixel contribution from each primitive - in effect generating a compound pixel. Similarly, step (b) may further comprise selecting per pixel contributions from two or more overlapping primitives, at least the nearest of which is non-opaque and both or all of which are behind the first opaque primitive.

30 Also in accordance with the present invention there is provided image

processing apparatus for generating a stereo pair of 2-D images of an object made up of a plurality of image primitives in 3-D object space, the first image being from a predetermined viewpoint within the object space, and the apparatus comprising a host processor with associated main memory for the storage of object primitive data and a display processor with associated display memories, the host processor comprising means for supplying object primitive data to the display processor, and the display processor comprising a first z-buffer sorting the primitive data to select the nearest opaque primitive per pixel of the first 2-D image, writing the selected pixel data to the first display memory, and outputting rejected pixel primitive data having a greater depth; a second z-buffer coupled to receive the output rejected pixel primitive data from the first z-buffer to select the nearest opaque primitive per pixel from the received data; positional offset means arranged to apply a positional offset in one of said two dimensions to the coordinates of pixel primitive data selected in the first and second z-buffers in an amount per pixel determined by the respective depth value of that pixel; and a third z-buffer coupled to receive the shifted pixel primitive data from the positional offset means to select preferential contributions per pixel, and write the selected pixel data to the second display memory.

The display apparatus host processor associated main memory may further hold object primitive texture definitions with the display processor including a texture memory. With such an arrangement the host processor may further comprise means for storing in the texture memory at least one pyramidal or part-pyramidal array of texel values comprising a plurality of two-dimensional (2-D) arrays of texel values representing a given 2-D modulation pattern at at least two levels of resolution defined by respective values of a level coordinate; and the above-mentioned means for supplying object primitive data to the first display processor may include an indication that a pattern of modulation is to be applied to the object primitive in accordance with texel values stored in the pyramidal array in the texture memory. Also, the display processor may further comprise means for generating from the object primitive

data a series of pixel addresses for application to at least the first display memory and a corresponding series of 2-D texture coordinate pairs each with an associated level coordinate, to effect a mapping of the stored modulation pattern onto the object primitive at a level or levels of resolution defined by the 5 associated level coordinate; means operable to generate an offset to each said associated level coordinate, and to generate said texture memory addresses from the received coordinate pair and the sum of the associated level coordinate and generated offset thereto; and an input for receiving a focal depth value, and said means operable to generate said offsets is arranged to 10 generate said offsets having a respective magnitude determined by the received focal depth value.

A multi-view display apparatus comprising a number of display apparatuses as described above may be assembled, the display memories of 15 which supply images to respective views of a multi-view lenticular array screen, with the respective images being visible at a number of horizontally displaced viewpoints in front of the screen, the views towards the centre of the horizontally displaced series being driven from the respective first display memories, and the outer views being driven by the respective ones of the second display memories.

20

Further features and advantages of the present invention will become apparent from reading of the following description of preferred embodiments of the present invention, given by way of example only, and with reference to the accompanying drawings, in which:

25 Figure 1 is a plan schematic view illustrating parallax shift effects between two horizontally spaced viewpoints;

Figure 2 shows three overlapping image primitives as seen from each of the viewpoints of Figure 1;

30 Figure 3 is a block diagram of a stereoscopic display apparatus including image rendering and texture mapping hardware and embodying the present invention;

Figure 4 schematically illustrates the depth cue circuit arrangement of the mapping hardware of Figure 3; and

Figure 5 represents different depth cueing characteristics which may be applied by the circuit of Figure 4.

5

As will become apparent from the following, the present invention is applicable to both conventional and screen space 3-D graphics renderers. A conventional renderer is one in which rendering primitives (typically triangles) are written sequentially to a frame buffer and, as such, any pixel of the final image may be written at any time. A screen space renderer splits the screen into smaller areas of  $M \times N$  pixels called tiles; this includes so-called scanline renderers where  $M$  is the width of the screen and  $N$  is 1 pixel. For each tile the screen space renderer determines which primitives contribute to (are overlapped by) that tile, performs rendering processes such as texturing, and writes pixel values for that tile to the frame buffer.

10 A conventional or screen space renderer can perform depth sorting for rendering primitives in each screen or tile using conventional z-buffer methods as described for example in the above-mentioned Foley et al reference at pages 668 to 672. The z-buffer algorithm is used to deduce the nearest visible rendering primitive at each pixel and hence the colour of the pixel to be output. The screen space renderer need only maintain a z-buffer for each tile, whereas a conventional renderer must maintain a z-buffer for the screen.

15 The technique to be described is equally applicable to both conventional and screen space renderers, but it is more appropriate to screen space renderers because of their smaller z-buffer requirements. Neither is the technique restricted to renderers using the conventional z-buffer; it is applicable to those renderers that accumulate some or all of the rendering primitives impacting each pixel in order that attributes of these rendering primitives can be combined (in some manner) to determine the final pixel colour and depth; this composite can be considered as the nearest opaque pixel information in 20 subsequent discussions.

Figure 1 is a plan view showing the parallax effect leading to the appearance of gaps in conventional techniques for a pair of viewing positions V1 and V2 which may suitably comprise a stereo pair. The screen on which the viewer sees the 2-D transformed version of the 3-D virtual environment is indicated at 10, with the area to the right of this representing apparent depth within the image field (the z-direction). Three image primitives are represented at 12, 14 and 16: for simplicity of illustration, it is assumed that all three are parallel and vertical planar bodies. The primitive 12 with the smallest depth value z (hence the closest to the viewer) is assumed to be translucent whilst the two further back primitives 14, 16 are each assumed to be opaque.

The view of the three primitives from the respective viewpoints V1 and V2 is shown in Figure 2. The images illustrate the problem where the image for the right eye (V1) is rendered complete and that for the left eye generated by post-processing, with a gap G appearing in the V2 image where the shift in the x-direction has uncovered an area of the primitive 16 for which, during z-buffering of the V1 image data, the data was discarded as obscured by the opaque surface of primitive 14. A further potential gap area is indicated at G' which is revealed by the parallax shift to the translucent primitive 12: whether or not a gap appears will depend on whether a compound value for the colour due to both primitives is held or whether the data for the two primitives 12 and 14 is separately stored for the G' area. Typically a renderer will maintain the depth of the nearest opaque primitive, although screen space renderers in particular might maintain related information such as the depths of translucent or partially covered primitives in front of the aforementioned opaque primitive. Renderers typically discard primitives such as 16 behind the nearest opaque primitive 14.

In order to overcome this defect, stereo image generating apparatus embodying the invention maintains information about the primitive that is immediately behind the nearest opaque primitive for each pixel; this is referred to hereinafter as the nearest occluded pixel information. In effect a z-buffer is maintained for not only the nearest visible pixels but also for the nearest

occluded pixels. Consequently, when pixels corresponding to the nearest opaque pixels are shifted to the left or right during generation of the synthetic stereo images (for example, the left hand edge of primitive 14), the nearest occluded pixel information (for the previously hidden part of primitive 16) is 5 available to fill in the gaps caused by shifting. The nearest occluded pixel information typically corresponds to the actual scene content, so the filled gaps are highly correlated to the remainder of the image and as such are unobtrusive.

Maintaining the occluded pixels is expensive (but not prohibitive) for a 10 conventional renderer because it must in effect maintain two full screen z-buffers and two full screen frame buffers for each frame. A screen space renderer may already be maintaining depth information for several rendering primitives per pixel, making maintenance of the nearest occluded pixel relatively inexpensive. The two sets of depth and colour information for each pixel need 15 only be held for the pixels in a given tile, which is considerably less expensive than the cost of maintaining two such sets for every pixel on the screen.

In terms of maintaining throughput for real time operation, a conventional 20 renderer must necessarily texture and illuminate the occluded pixels so there is no loss in throughput there, but the z-buffer depth sorting done at the frame buffer writing stage must be done twice as fast as normal. A screen space renderer typically does depth sorting before texturing and illumination, so it must double its throughput during and after depth sorting, however it still need not texture or illuminate otherwise occluded pixels.

Figure 3 is a block diagram of a scanline stereoscopic rendering and 25 display apparatus. A keyboard 21 and a trackball input device 23 provide input from a user to a central processing unit (CPU) 25, other input devices may also be used, of course, such as a joystick, digitising tablet, or a mouse. Objects and also photographic images to be applied to object surfaces by texture mapping may also be input from a video source such as a camera 19.

The CPU 25 is connected via a bus 18 to a disc store 20, a ROM 22 and 30 a main memory (MRAM) 24. The disc store, which may include magnetic

floppy discs, hard discs, and/or optical memory discs, is used for storing data (for example images or 3-D model data) which can then be recalled and manipulated to generate new images as desired. Such data may include the user's work from previous input sessions and/or commercially generated data, 5 for example for use in interactive computer-aided design or computer simulations for education or entertainment. To allow modelling of 3-D objects, such data will generally be stored as polygonal model data rather than in the form of two-dimensional images. In that case, the data corresponds to a 3-D model containing objects which are typically broken down into groups of 10 polygonal surfaces (primitives) in 3-D object space. The data for each object in the model comprises a list giving the position and nature of every polygon that goes to make up the object, including the relative positions of its vertices and the colour or transparency of the polygon surface. In other systems, primitives may comprise curved surface patches, as is known in the art. It is 15 known that a texture can be specified for mapping onto the surface, so that detail can be represented without increasing the number of primitives that make up the scene. A texture map is a stored 2-D array of texture element (texel) values defining a 2-D pattern of modulation that may for example define the colour of pixels in a manner to be described below, or may modulate other 20 quantities such as reflectance or surface normal direction. These texture maps may also be stored in the disc store 20 and recalled as required.

The CPU 25 and the other components of the system then translate the 3-D model "world" in object space into a first two-dimensional view for the user (in "viewer" space), from a first viewpoint which may be determined by the 25 application or may be controlled by a user. As will be described below, the second view is generated by post-processing, with the first and second views forming a stereo pair for display on an autostereoscopic display screen (not shown) or a VR-type head-mounted display (HMD) 34.

The translation is effected by means of geometric transformations 30 effecting translations, rotations and perspective projections, generally by means matrix multiplication of vertex coordinates, and the CPU 25 may also perform

clipping and lighting calculations on a per-primitive or per-vertex basis. The ROM 22 and MRAM 24 provide program memory and workspace for the CPU 25, and special processing hardware 26 may be provided to assist the CPU 25 to perform the large number of arithmetic operations required to convert all but 5 the simplest models into a two-dimensional scene. The hardware 26 may comprise standard arithmetic circuits or it may include more powerful custom-built or programmable digital signal processing integrated circuits, and may be connected to the CPU 25 via bus 18. The nature of the hardware 26 will depend on the requirements of the system, for example with respect to speed, 10 resolution, number of primitives per scene, etc.

A display processing unit (DPU) 28 is connected between outputs of the CPU 25 (via the bus 18) and inputs of a first display memory (VRAML) 30 for the left hand image of the pair. The display memory 30 stores pixel data COLL in raster-scan format. The pixel data COLL might typically include for each 15 pixel three 8-bit values (total 24 bits) corresponding to red (R) green (G) and blue (B) components of the desired image. Those skilled in the art will appreciate that in other embodiments fewer or more bits may be provided for, or the bits might define the colour in terms of different components.

In the DPU 28 the primitives are "scan converted" so that the final left 20 hand image may be drawn a line at a time into the left hand display memory 30. Scan conversion is a process whereby the pixels covered by each primitive are written row by row and pixel by pixel, in the same way that the complete image will be scanned for output to the display.

The DPU 28 is coupled to first 60, second 62 and third 64 scanline 25 buffers, with the first 60 performing hidden surface removal based on the relative z-value of each primitive contributing to a pixel. As a primitive pixel is "covered", which is to say an opaque pixel having a smaller depth is written into its place in the first buffer 60, the displaced primitive pixel is copied to the corresponding position in the second buffer 62. When all primitives 30 corresponding to that scanline have been considered, the pixel values in buffer 60 are simply read out to the corresponding line in VRAML 30. The pixel

values in buffer 62 represent the value for the first occluded primitive at each pixel location.

To generate the right hand image of the pair, a shift in the x direction is applied to pixel values in buffers 60 and 62 in an amount determined by the 5 respective depth values of the individual pixels and these values are then entered in the buffer 64 at the new locations. As before, a value can only be entered if there is not already a nearer opaque pixel at that location so that majority of values from buffer 62 will be rejected in favour of those from buffer 60 but, where the parallax shift has introduced a gap (as in Figure 2) the 10 necessary pixel values for the first occluded primitive will generally be present from buffer 62.

When all values have been written into buffer 64, the resulting contents comprise the correct pixel values for the corresponding line in the shifted right hand image and the data is written directly to a second display memory 66 (VRAMR). 15

A timing unit (video controller) 32 generates read-address signals XD and YD to address the pixel data within the two display memories VRAML 30 and VRAMR 66 synchronously with the raster-scanning of respective display screens in HMD 34. In response to these address signals, the locations in 20 VRAML 30 and VRAMR 66 are scanned row by row and column by column to read respective colour values COLDL and COLDR which are fed to display converters 36, 68 which translate the pixel data COLDL and COLDR into the appropriate form of signal for supply to the HMD 34, which also receives timing signals (SYNC) from the timing unit 32. As will be readily understood, the form 25 and/or operation of components 30,32,36,66,68 may vary for different display devices, for example for an interlaced autostereoscopic display having an overlaid lenticular screen.

The pixel values COL are generated so that a basic surface colour of the primitive is modulated to account realistically for attributes of an object's surface (for example transparency, diffuse reflectance, specular reflectance) 30 and of the 3-D environment (for example locations and shapes of light sources).

Some of this modulation can be generated arithmetically from parameters loaded with the primitive data, for example to produce smoothly varying shading to simulate a curved surface. However, to provide more detailed modulation, and to facilitate the introduction of depth cueing to the image, 5 mapping hardware is provided to supply modulation values MOD based on a predetermined pattern stored in advance in a texture memory 41.

To this end, the DPU 28 generates a pair of texture coordinates U and V for each pair of pixel (display) coordinates X and Y so that the modulation pattern is mapped onto the primitive surface, implementing geometric 10 transformations (i) from texture space into object space and (ii) from object space into viewer (display) space. Note that the texturing is done after sorting by the z buffers 60,62 to avoid the waste of rendering primitive areas which are definitely occluded.

The texture coordinates U and V are processed within the mapping 15 hardware in a manner to be described below and applied to the texture memory 41 so that a modulation value MOD is available for each display pixel location X,Y being addressed. Commonly, even if the values MOD are colour values, they will require to be modified within the DPU 28 to allow for realistic lighting 20 effects. In a more general case, the modulation values MOD are used within the DPU 28 together with other parameters to modify the pixel values COLL and COLR less directly.

The texels represented in the texture memory 41 will not in general correspond on a one-to-one basis with the pixels of the display and, in particular when the primitive is shown in the distance and the texture is 25 consequently mapped onto a very small number of pixels, two-dimensional spatial filtering is required to avoid the aliasing effects that would occur if simple sub-sampling were used.

It is known that a generalised filter cannot be applied economically in an apparatus where real-time moving images are to be synthesised, and the 30 conventional solution to this is to store several 2-D arrays (hereinafter referred to as "maps") for a given pattern, each being successively smaller and pre-

5 filtered to a successively lower resolution. The DPU 28 then need only produce a level coordinate  $L$  to determine the appropriate map to use. For compact storage and for high speed access to the texel values, the maps may be chosen to be square, having power-of-two dimensions, and be stored in a square texture memory according to the so-called "multum in parvo" ("MIP map") technique.

10 Figure 3 shows within the texture memory 41 the colour components R, G and B of a texture pyramid stored as a MIP map. The largest (highest resolution) map ( $L=0$ ) may for example comprise  $512 \times 512$  texels, the  $L=1$  maps comprise  $256 \times 256$  texels and so on down to  $L=9$  where each map becomes a single texel. Assuming, for the sake of example, that each texel value comprises an 8-bit value for each of the R, G and B colour components, the entire texture memory 41 is thus 1 Mbyte in size.

15 The texel values are stored in the memory 41 in advance of rendering by the CPU 25 via the bus 18 and a writing port 43 of the memory 41. For each texel value to be read, the DPU 28 generates a 2-D coordinate pair, each coordinate (U and V) of which includes at least an integer part 9 bits in length. At the same time, the level coordinate  $L$  is generated by the DPU 28 and, subject to modification by a depth cue circuit 40, is used to generate physical 20 coordinates  $U'$  and  $V'$  from the "virtual" coordinates U and V for application to read address ports 44 and 45 respectively of the texture memory 41. In response to each physical coordinate pair  $U', V'$ , the memory 41 releases the R, G and B components of an addressed texel via a (24-bit) read port 46.

25 Because of the two-dimensional binary tree arrangement of the MIP maps in the memory 41, the required physical coordinates  $U'$  and  $V'$  can be generated simply by a pair of binary shifting circuits 47 and 48 respectively, each right-shifting the respective coordinate a number of places defined by the level coordinate  $L$ . In particular, if  $L=0$  represents the highest level, then the address corresponding to a given texel in the level 0 map can be converted to the physical address of the corresponding texel in the level  $L$  map can be found 30 by right-shifting the U and V coordinates  $L$  places, effectively scaling-down

each coordinate by  $2L$ . The unmodified level coordinate  $L$  can be supplied to the DPU 28 as part of the primitive data, but if perspective is to be accounted for in the mapping, then the level coordinate  $L$  will more probably be generated within the DPU on a per-pixel basis, dependent on the partial derivatives of  $U, V$  with respect to  $X, Y$ .

5 The depth cue circuit 40 is shown in Figure 3 as a separate unit coupled to the DPU 28 although, as will become apparent, its functions may equally be performed in software within the DPU. As shown schematically in Figure 4, the circuit 40 has inputs for the unmodified level coordinate  $L$ , a focus depth  $F$  and a pixel depth  $z$ . The focus depth  $F$  specifies that apparent depth within the 2-D display image at which the viewer is intended to focus. The value of  $F$  may be 10 fixed, or may be variable under control of the application or in response to user input, as required by the particular application. The pixel depth is supplied for each pixel in the displayed image, in synchronism with the generated display 15 coordinates  $X$  and  $Y$  and is generated by the DPU 28 during the translation of the 3-D world model to the 2-D display image.

An arithmetic stage 70 generates an output value  $FS$  as a function of the separation between the pixel depth and focus depth which value is added to the unmodified level coordinate  $L$  at summation circuit 72 to produce a modified 20 coordinate level  $L'$ . A limiter stage 74 is suitably provided on the output of the summation circuit to ensure that the modified coordinate  $L'$  is within the range of levels supported by the texture memory 41. The particular function applied by arithmetic circuit 70 may vary depending on the desired depth cue characteristic, as shown by examples A, B and C in the graph of  $FS$  against 25 separation ( $F-z$ ) in Figure 5. The common feature is that for pixel depths at or close to the focus depth  $F$ , the value of  $FS$  is zero such that  $L'=L$  and those pixels have the "appropriate" texture resolution whilst the texture is blurred at other depths.

Where the mapping hardware includes interpolators (not shown) for 30 generating texture values by interpolating between two map levels of the pyramidal data structure, non-integer values of  $L'$  may be allowed, as shown

by examples A and B (which also show the effect of the limiter stage 74 when the maximum value of L for the texture memory is reached). Where only integer values of L (and hence L') are supported, FS is similarly constrained to be an integer value as shown by example C.

5 In summary, we have described a post-processing method and apparatus whereby a second image is formed by horizontally shifting pixel primitives of a first image by an amount determined by their respective depths to give a parallax shift. To avoid gaps appearing in the shifted image, two z-buffers are maintained during rendering of the first image with the second being  
10 loaded by discarded values of the first such as to provide a record of the first occluded primitive for each pixel. The horizontal shift is applied to the contents of both z-buffers and the results written to a third z-buffer such that gaps due to the varied shift applied to the first image values (the unshifted contents of the first buffer) are filled by previously occluded values from the second buffer.  
15 The two images may then be output to a suitable stereoscopic display such as a Virtual Reality headset.

From reading the present disclosure, other modifications will be apparent to persons skilled in the art. Such modifications may involve other features which are already known in the field of autostereoscopic display apparatus and  
20 component parts thereof and which may be used instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure of the present application also includes any novel feature or any novel combination of features disclosed herein either  
25 explicitly or implicitly or any generalisation thereof, whether or not it relates to the same invention as presently claimed in any claim and whether or not it mitigates any or all of the same technical problems as does the present invention. The applicants hereby give notice that new claims may be formulated to such features and/or combinations of such features during the prosecution of the present application or of any further application derived  
30 therefrom.

## CLAIMS

1. A method for generating a stereo pair of two-dimensional (2-D) images of an object made up of a plurality of image primitives in three-dimensional (3-D) object space, the first image being from a predetermined viewpoint within the object space and the method comprising the steps of:
  - a) z-buffering the primitive data to select the nearest opaque primitive per pixel of the first 2-D image and rejecting pixel primitive data having a greater depth;
  - 10 b) z-buffering the rejected pixel primitive data from step (a) to select the first occluded primitive per pixel of the first 2-D image;
  - c) applying a positional offset in one of said two dimensions to the pixel primitive data selected in steps (a) and (b) in an amount per pixel determined by the respective depth value of that pixel;
  - 15 d) z-buffering the shifted pixel primitive data from step (c) to select preferential contributions per pixel; and
  - e) rendering the first image from the pixel primitive data selected at step (a) and the second from the pixel primitive data selected at step (d).
- 20 2. A method as claimed in Claim 1, wherein step (a) further comprises selecting non-opaque primitives if nearer than said nearest opaque primitive per pixel, and generating a per pixel contribution from each primitive.
- 25 3. A method as claimed in Claim 1, wherein step (b) further comprises selecting per pixel contributions from two or more overlapping primitives, at least the nearest of which is non-opaque and both or all of which are behind the first opaque primitive.
- 30 4. A method as claimed in Claim 1, wherein step (c) comprises applying a horizontal positional offset.

5. Image processing apparatus for generating a stereo pair of two-dimensional (2-D) images of an object made up of a plurality of image primitives in three-dimensional (3-D) object space, the first image being from a predetermined viewpoint within the object space and the apparatus comprising a host processor with associated main memory for the storage of object primitive data and a display processor with associated display memories, the host processor comprising means for supplying object primitive data to the display processor, and the display processor comprising:

10 a first z-buffer sorting the primitive data to select the nearest opaque primitive per pixel of the first 2-D image, writing the selected pixel data to the first display memory, and outputting rejected pixel primitive data having a greater depth;

15 a second z-buffer coupled to receive the output rejected pixel primitive data from the first z-buffer and to select the nearest opaque primitive per pixel from the received data;

20 positional offset means arranged to apply a positional offset in one of said two dimensions to the coordinates of pixel primitive data selected in the first and second z-buffers in an amount per pixel determined by the respective depth value of that pixel; and

25 a third z-buffer coupled to receive the shifted pixel primitive data from the positional offset means to select preferential contributions per pixel, and write the selected pixel data to the second display memory.

30 6. A display apparatus as claimed in Claim 5 wherein said host processor associated main memory further holds object primitive texture definitions and the display processor includes a texture memory;

35 the host processor further comprising means for storing in the texture memory at least one pyramidal or part-pyramidal array of texel values comprising a plurality of 2-D arrays of texel values representing a given 2-D modulation pattern at at least two levels of resolution defined by respective values of a level coordinate; and

said means for supplying object primitive data to the first display processor, including an indication that a pattern of modulation is to be applied to the object primitive in accordance with texel values stored in the pyramidal array in the texture memory; and

5 the display processor further comprising means for generating from the object primitive data a series of pixel addresses for application to at least the first display memory and a corresponding series of 2-D texture coordinate pairs each with an associated level coordinate, to effect a mapping of the stored modulation pattern onto the object primitive at a level or levels of resolution 10 defined by the associated level coordinate; means operable to generate an offset to each said associated level coordinate, and to generate said texture memory addresses from the received coordinate pair and the sum of the associated level coordinate and generated offset thereto; and

15 an input for receiving a focal depth value, and said means operable to generate said offsets is arranged to generate said offsets having a respective magnitude determined by the received focal depth value.

7. A display apparatus as claimed in Claim 6, wherein said second display processor has an associated texture memory corresponding to that of 20 said first display processor, together with a respective input for receiving said focal depth value and means operable to generate said offsets.

8. A display apparatus as claimed in Claim 6, further comprising user 25 operable input means coupled to said host processor, said host processor being arranged to generate and vary said focal depth value in dependence on signals from said input means and to output said focal depth value to the or each display processor.

9. A multi-view display apparatus comprising (N+1) display 30 apparatuses according to Claim 5, the display memories of which supply images to respective views of a 2(N+1) view lenticular array screen, with the

respective images being visible at  $2(N+1)$  horizontally displaced viewpoints in front of the screen, the  $(N+1)$  central views of the horizontally displaced series being driven from the  $(N+1)$  first display memories, and the outer views being driven by the respective ones of the second display memories, where  $N$  is an integer and  $N=(1,2,3 \dots \text{etc})$ .

1/3

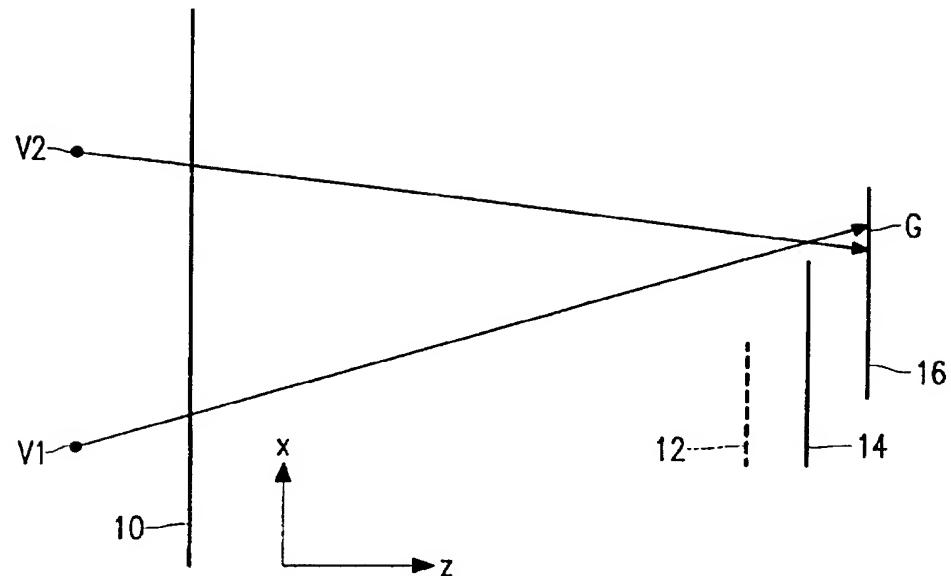


FIG. 1

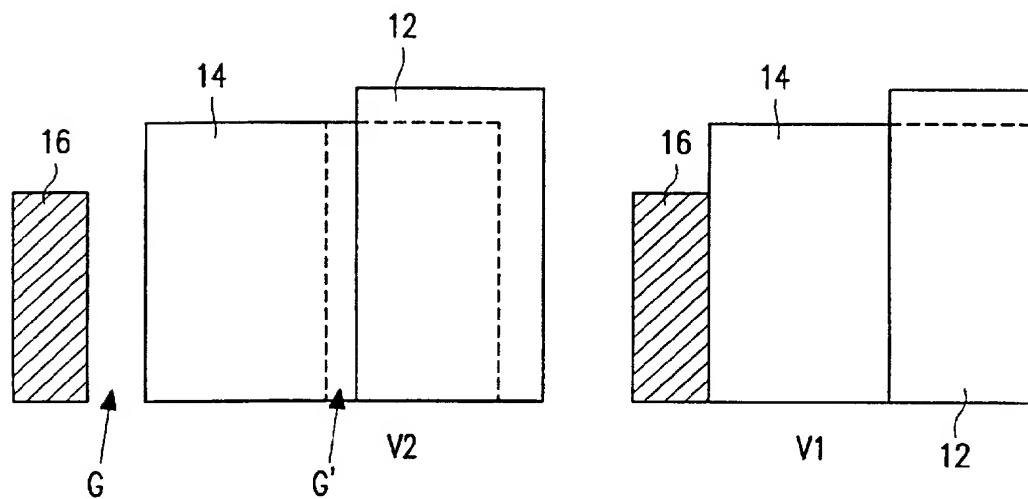


FIG. 2

2/3

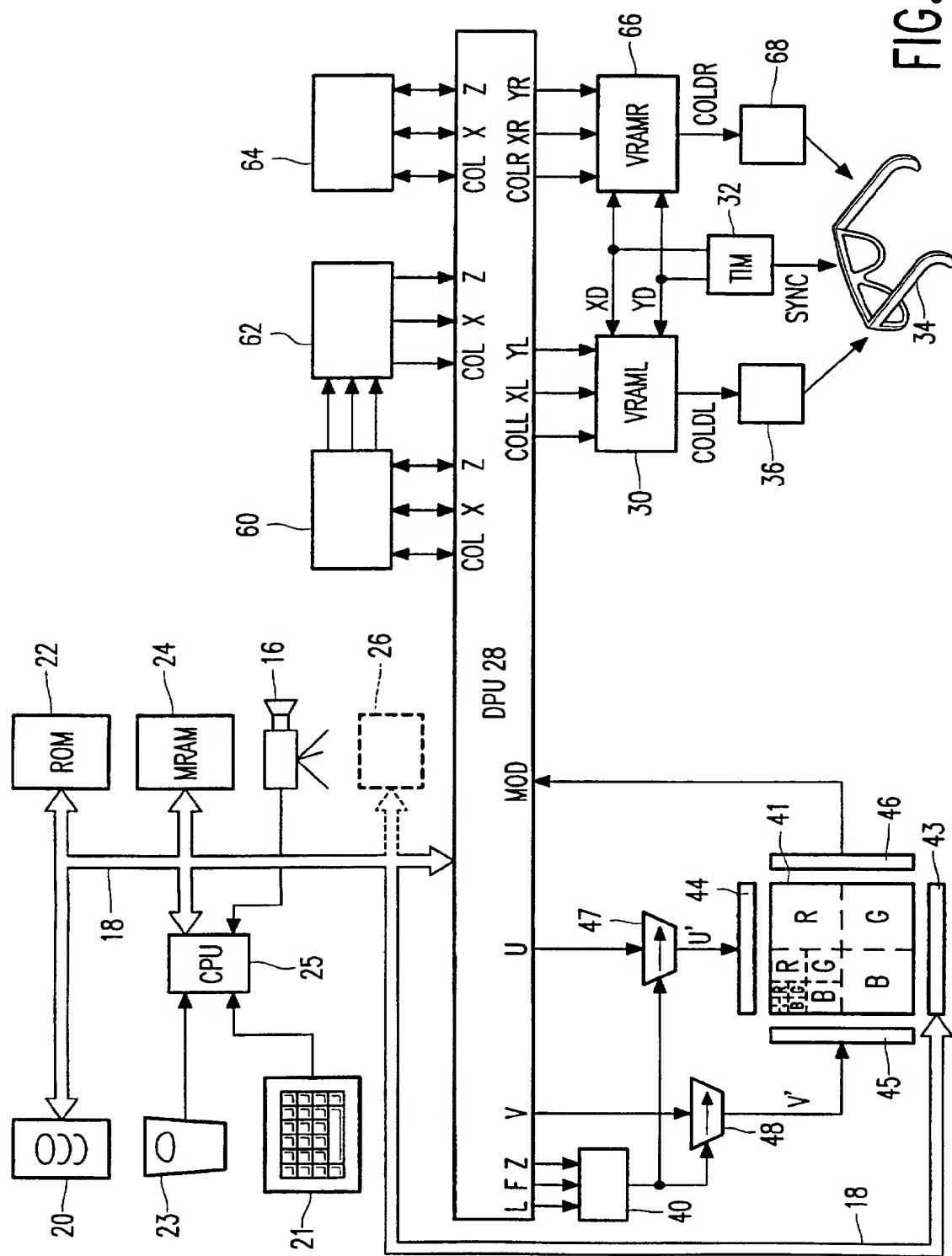


FIG. 3

3/3

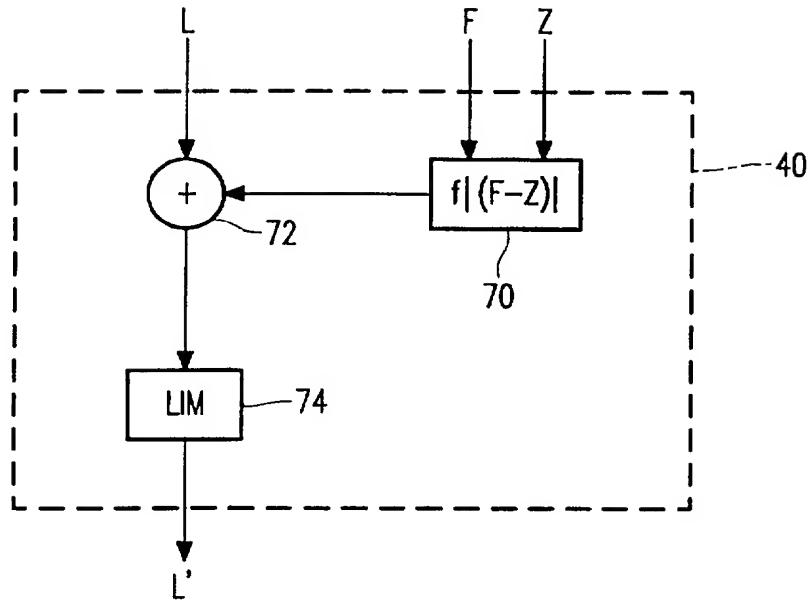


FIG. 4

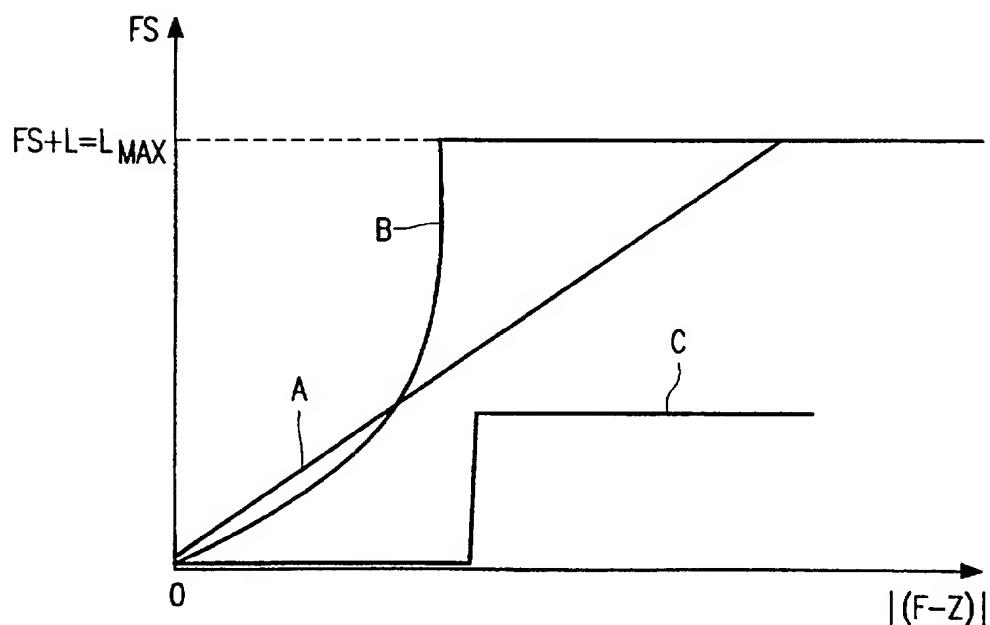


FIG. 5